NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

TRELLIS CODED MODULATION APPLIED TO
ORTHOGONAL SIGNALS

by

Janet Emerson Stevens

March 1993

Thesis Advisor: Tri T. Ha

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 1993

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
TRELLIS CODED MODULATION APPLIED TO ORTHOGONAL SIGNALS

5. FUNDING NUMBERS

6. AUTHOR(S)
Stevens, Janet E.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the US Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A method for the design of trellis codes for coherent detection of orthogonal signals in *additive white Gaussian noise* (AWGN) channels is presented. This method utilizes the landmark works of Ungerboeck in *trellis coded modulation* (TCM). After examining the channel capacity, it is shown that a coding method requiring the same bandwidth efficiency for the orthogonal signal space and *maximum likelihood* (ML) soft decoding using the Viterbi algorithm can achieve large asymptotic coding gains. Several codes are analyzed using Ungerboeck's technique of set-partitioning and mapping, then applying the analytic code description method of Calderbank and Mazo to *M-ary frequency shift keying* (M-FSK). The general finding of this paper is that relative to uncoded modulation, asymptotic coding gains of 3-4 dB can be achieved.

14. SUBJECT TERMS
trellis coded modulation, orthogonal signals, M-ary frequency shift keying, Viterbi algorithm

15. NUMBER OF PAGES
62

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std Z39-18

Approved for public release; distribution is unlimited.

Trellis Coded Modulation Applied to Orthogonal Signals

by

Janet Emerson Stevens
Lieutenant, United States Coast Guard
B.S.E.E., United States Coast Guard Academy, 1988

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1993

Author: Janet E. Stevens
Janet E. Stevens

Approved by: Tri T. Ha
Tri T. Ha, Thesis Advisor

Ralph Hippenstiel
Ralph Hippenstiel, Second Reader

Michael A. Morgan
Michael A. Morgan, Chairman,
Department of Electrical and Computer Engineering

ABSTRACT

A method for the design of trellis codes for coherent detection of orthogonal signals in *additive white Gaussian noise* (AWGN) channels is presented. This method utilizes the landmark works of Ungerboeck in *trellis coded modulation* (TCM). After examining the channel capacity, it is shown that a coding method requiring the same bandwidth efficiency for the orthogonal signal space and *maximum likelihood* (ML) soft decoding using the Viterbi algorithm can achieve large asymptotic coding gains. Several codes are analyzed using Ungerboeck's technique of set-partitioning and mapping, then applying the analytic code description method of Calderbank and Mazo to *M-ary frequency shift keying* (M-FSK). The general finding of this paper is that relative to uncoded modulation, asymptotic coding gains of 3-4 dB can be achieved.

TABLE OF CONTENTS

ACKNOWLEDGMENT

I wish to thank Professor Ha and Professor Hippenstiel for their time, instruction, patience, and support in this thesis. Also, I appreciative my husband, Jim, and my boys, Scott and Todd, who always knew how to make my day. Finally, I thank Cessna for keeping Jim company, and JFK for keeping my adrenaline flowing.

I. INTRODUCTION

Forward error correction coding is often employed in military communications systems to improve their performance in a hostile environment. Both block coding and convolutional coding are commonly used. An alternative coding technique is *trellis coded modulation* (TCM). In this thesis M-ary orthogonal signals are considered.

The model of the digital communication system under consideration is shown in Figure 1.1. The trellis code encodes a binary data stream as a sequence of signal points drawn from M-dimensional Euclidean space R^M. At a given instant in time, the input to the convolutional encoder, $\{a_i\}$, is a sequence of k independent bits where a_i=0,1 and i=1,...,k. The channel encoding operation is combined with *M-ary frequency shift keyed* (M-FSK) modulation, thus trellis coded modulation. A convolutional encoder is used to generate

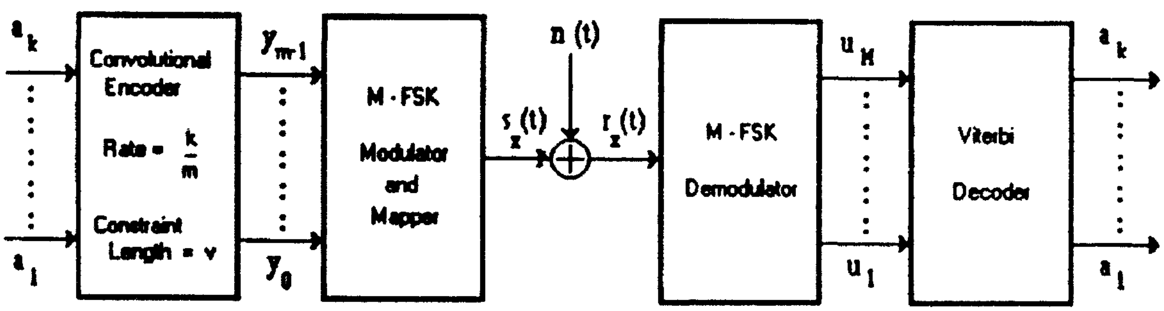


Figure 1.1 Communication System Model

the underlying code. M-FSK orthogonal signaling is used to achieve an acceptable performance with a minimum power requirement. [Ref. 1]

The set of orthogonal signals used is denoted as

$$\{s_x(t)\}, \; x = 1,2,\ldots,M. \tag{1.1}$$

These signals are generated by a frequency synthesizer which is driven by a set of real numbers $\{x\}$, $x = 1,2,\ldots,M$. So, each x corresponds to one of the M transmitted frequencies. These real numbers have a specific relationship, denoted by a generator matrix or polynomials, with the incoming bit stream $\{a_i\}$ and the memory bits of the encoder. When a trellis code is used to encode data at the rate k bits/channel symbol, each channel input $\{y_j\}$, $j=0,\ldots,m-1$ ($m=\log_2 M$), depends on the most recent block of k bits to enter the encoder and the set of v bits preceding this block. Here v is equal to the number of memory elements in the encoder, or the number of shift register stages. These v bits determine the state of the encoder and the most recent block of k bits generates the channel symbol conditional on the encoder state. For an encoder that has 2^v states, the constraint length of the convolutional code is defined as v.

The $M=2^m$ distinct symbols are represented by M orthogonal sinusoidal waveforms as

$$s_x(t) = A \cos\,[2\pi(f_c+(x-1)\Delta f)t], \qquad 0 \le t \le T \qquad (1.2)$$

where x = 1,2,...,M,

T : symbol duration,

f_c : carrier frequency,

Δf : the width of the frequency slot.

The orthogonal signal set is characterized by equal signal energy E given by

$$E = \int_0^T s_x^2(t)\,dt = A^2\,T/2. \qquad (1.3)$$

Utilizing (1.3), we may equivalently represent these signal waveforms as M-dimensional vectors:

$$\begin{aligned} \mathbf{s}_1 &= [\sqrt{E},\ 0,\ \ldots,\ 0] \qquad (1.4)\\ \mathbf{s}_2 &= [\ 0,\ \sqrt{E},\ \ldots,\ 0]\\ &\ \vdots\\ \mathbf{s}_M &= [\ 0,\ 0,\ \ldots,\ \sqrt{E}\ ]. \end{aligned}$$

Euclidean distance, much like the cross-correlation coefficient, is a measure of the similarity or dissimilarity between any pair in the set of signal waveforms or

corresponding signal vectors. [Ref. 2] The distance between two vectors **f** and **g** is defined as

$$d(f,g)=\sqrt{\sum_{n=1}^{\infty}|f_n-g_n|^2} \tag{1.5}$$

where f_n and g_n are the elements of **f** and **g**, respectively. Using the above notation, it easy to see that the distance between any two signals $\mathbf{s}_i$ and $\mathbf{s}_j$, $i\neq j$, is

$$d=\sqrt{2E}. \tag{1.6}$$

For these signals, the cross-correlation coefficient obeys the relationship

$$\rho_{ij} = \int_0^T s_i(t)s_j(t)\ dt = \begin{cases} 0 \text{ for } i\neq j \\ E \text{ for } i=j. \end{cases} \tag{1.7}$$

M-FSK is generated by subdividing a frequency interval into M distinct frequency slots. Each slot has a width Δf. The binary digits output from the convolutional encoder, are mapped into a set of M symbols corresponding to the frequency slots. From Reference 2, we see that the minimum frequency separation between adjacent signals that uphold conditions for orthogonality is

$$\Delta f_{min}=\frac{1}{2T}. \tag{1.8}$$

This is precisely the separation used in coherent detection. In some systems, noncoherent detection is used. It is, however, half as bandwidth efficient as coherent detection. The frequency separation in this case is

$$\Delta f_{min} = \frac{1}{T}. \tag{1.9}$$

For coherent detection, the minimum one-sided bandwidth occupied by these M orthogonal signals is

$$B = \frac{M}{2T}. \tag{1.10}$$

It is assumed that the transmission medium introduces zero mean, *additive white Gaussian noise* (AWGN) n(t) with power spectral density No/2. The received signal is corrupted by AWGN given by

$$r_x(t) = s_x(t) + n(t). \tag{1.11}$$

The purpose of coding is to gain immunity over this noise beyond that provided by standard uncoded transmission at the same data rate [Ref. 3].

The signal $r_x(t)$ is demodulated by a bank of either matched filters or correlators. When the initial phase can be estimated by the receiver, the demodulation is coherent, otherwise it is non-coherent. In this thesis, only coherent detected TCM/M-FSK schemes are considered.

The demodulator computes the metrics which are the Euclidean distances between each possible signal and the received signal. The outputs $\{u_k\}$, $k = 1,2,\ldots,M$ of the demodulator are then sent to the Viterbi decoder which performs a soft decision decoding of the M-FSK signal. The decoder is cognizant of the trellis structure and signal assignment of the encoder. The sequence $\{\hat{a}_i\}$ at the output of the Viterbi decoder is the maximum-likelihood estimate of the input sequence $\{a_i\}$. Ideally, $\{\hat{a}_i\}$ should be equivalent to $\{a_i\}$.

The remaining chapters in this thesis are organized as follows. Chapter II contains general background information about digital communications, trellis coded modulation, and the Viterbi decoder. In Chapter III the encoder design procedure is presented. Heuristic code design and verification by hand is infeasible for codes with large constraint lengths. Optimum codes must be found using computers. Specifically, an efficient method to determine free distance is needed as it relates directly to the code performance [Ref. 4]. Along this line, Chapter IV contains several example rate 1/2 and rate 2/3 encoder with their respective computer aided design results. A simulation program provides verification of the decodability of the rate 1/2 encoders by decoding without added noise. One may then simulate the code in the presence of noise.

II. GENERAL INFORMATION

A. BACKGROUND INFORMATION

Trellis coded modulation was invented as a method "to improve the noise immunity of digital transmission systems without bandwidth expansion or reduction of data rate." [Ref. 4] Shannon enumerated the limitation to transmission over a noisy channel by a quantity called channel capacity. Capacity of a noisy channel is the largest rate at which information can be transmitted reliably. In other words, if the data source rate is less than the channel capacity, proper encoding and decoding techniques enable us to communicate over a noisy channel with any arbitrary error rate. Otherwise, reliable communication is not possible.

In Reference 2, Proakis notes that the set of orthogonal waveforms achieves the channel capacity bound as $M \rightarrow \infty$ when the information rate $R < C_{\infty}$, where C_{∞} is the capacity of the infinite bandwidth AWGN channel. Transmitting more signals using M-FSK modulation requires more bandwidth, as noted in (1.10). It also requires a higher degree of complexity of the demodulator, that is, more matched filters or correlators are used. The advantage of increasing M "is a reduction in the *signal to noise ratio* (SNR) per bit required to obtain a specified probability of error." Ungerboeck observes that in TCM "limited distance growth and increasing numbers of nearest

neighbors" prevent coding gains from achieving the limit set by channel capacity [Ref. 4].

In the classical approach to channel encoding, the two functions of coding and modulation are regarded as separate operations. The modulator, channel, demodulator, and hard quantizer are cascaded. Here, the code design is "to exploit redundancy at the bit level to maximize the minimum Hamming distance between codewords [Ref. 5]." In other words, the ability to detect and/or correct errors can only be provided by the transmission of redundant bits, thus lowering the effective information rate per transmission bandwidth. In addition, hard amplitude or frequency decisions made in the demodulator prior to final decoding cause an "irreversible loss of information." [Ref. 6]

The idea of combining channel encoding and modulation to achieve coding gains is the basis of TCM schemes. The receiver performs a maximum likelihood soft decoding of the unquantized demodulator output, thus avoiding loss of information prior to decoding. The objective of the code design is "the maximization of the minimum Euclidean distance between encoded sequences" [Ref. 5].

In this thesis, channel encoding and M-FSK modulation are combined in order to take advantage of soft decision decoding. The code rate of the convolutional encoder is the ratio of the number of input bits to the number of output bits. It is

written as r=k/m, where k is the number of inputs and m is the number of outputs of the encoder. Here we restrict our discussion to codes where m is equal to k+1. For a given number of memory elements v, the trellis structure depends on the way this memory is distributed among the incoming k bit streams. (For this reason, the general program for creating the trellis for rate 2/3 is not feasible.)

For every k information bits, the rate k/m trellis encoder produces m coded bits $y_0, y_1, \ldots, y_{m-1}$. These bits are mapped to a unique member of the 2^m signal constellation. In this thesis, a mapping rule known as the natural mapping rule will be assumed for its simplicity and illustrative purposes. The signal label will be equal to the numerical value of the binary coded digits. For example, if $(y_2\ y_1\ y_0) = (1\ 1\ 0)$, then the signal label is 6. This label corresponds to the signal value x = 7 in (1.2). More details of the mapper will be provided with the design examples.

Each transmitted signal $\mathbf{s}_x$ at multiples of time T is a nonlinear function of the state of the encoder and the k information bits at its input.

Consider the convolutional encoder depicted in Figure 2.1. In this case, the generator polynomials are known. Here, $y_0 = a_1 \oplus a_3$ and $y1 = a_1 \oplus a_2 \oplus a_3$, where $\oplus$ is the modulo 2 sum. Drawing this encoding procedure sequentially in time results in a trellis structure. The trellis for this rate 1/2 code with k=1 and v=2 is shown in Figure 2.2. For example, if the

$k = 1, \quad v = 2$

$y_1 = a_1 \oplus a_2 \oplus a_3$

$y_0 = a_1 \oplus a_3$

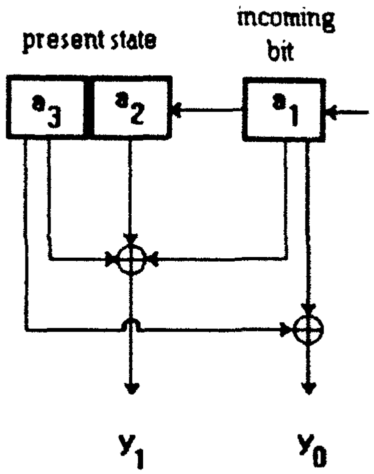


Figure 2.1 Inputs, state variables, and encoder connections for rate 1/2 code

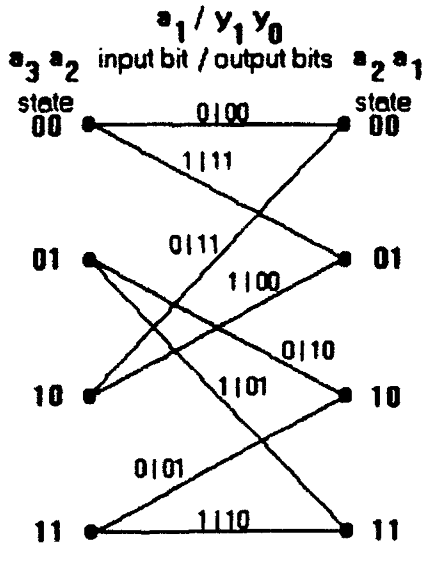


Figure 2.2 Trellis for rate 1/2 code

encoder is in state a_3a_2=00 and the new input bit a_1=1, then a transition from state 00 to state a_2a_1=01 occurs and the encoder output is y_1y_0=11. If the new input bit is a_1=0, then a transition from state 00 to state 00 occurs and the encoder output is y_1y_0=00. Other self-evident transitions are presented in Table 2.1. When a natural mapper is used, the trellis of Figure 2.2 is equivalent to the trellis of Figure 2.3. Here all states are numbered and transition branches are labeled with appropriate input/output signals.

Table 2.1 Trellis construction table.

From State a_3 a_2	Input Bit a_1	Output Bits* MSB LSB y_1 y_0	To State a_2 a_1
0 0	0	0 0	0 0
	1	1 1	0 1
0 1	0	1 0	1 0
	1	0 1	1 1
1 0	0	1 1	0 0
	1	0 0	0 1
1 1	0	0 1	1 0
	1	1 0	1 1

* MSB : Most significant bit
LSB : Least significant bit

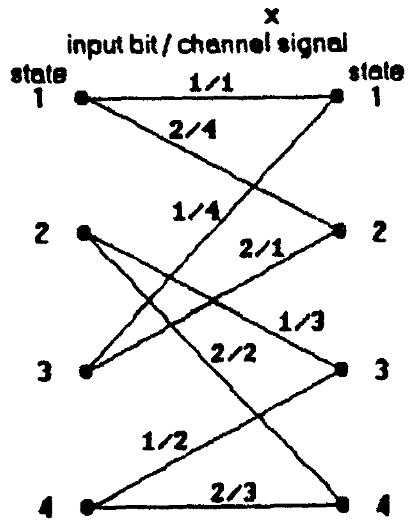


Figure 2.3 Signal assignment for rate 1/2 code

The 4-FSK and 8-FSK signal sets are presented in the top portions of Figures 2.4 and 2.5. The numbers below the line are the signal labels associated with the natural mapper. The values in the rectangles are the associated channel signal values x from (1.2). Referring to the bottom portion of Figures 2.4 and 2.5, the signal set is divided until each

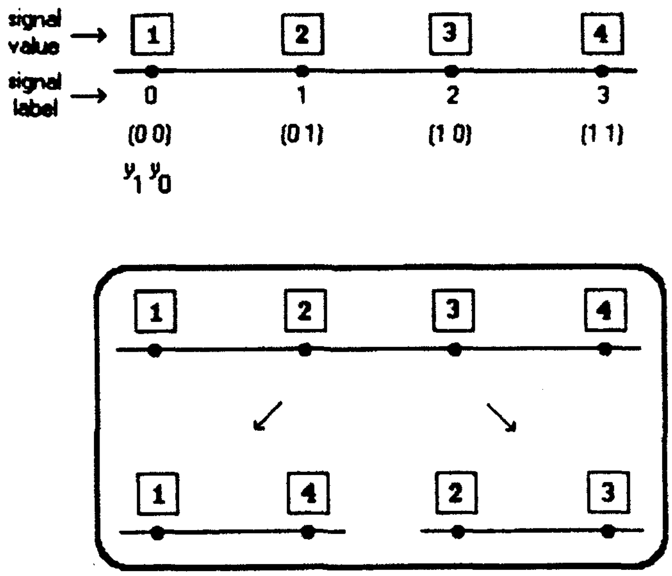


Figure 2.4 4-FSK signal set and set partitioning

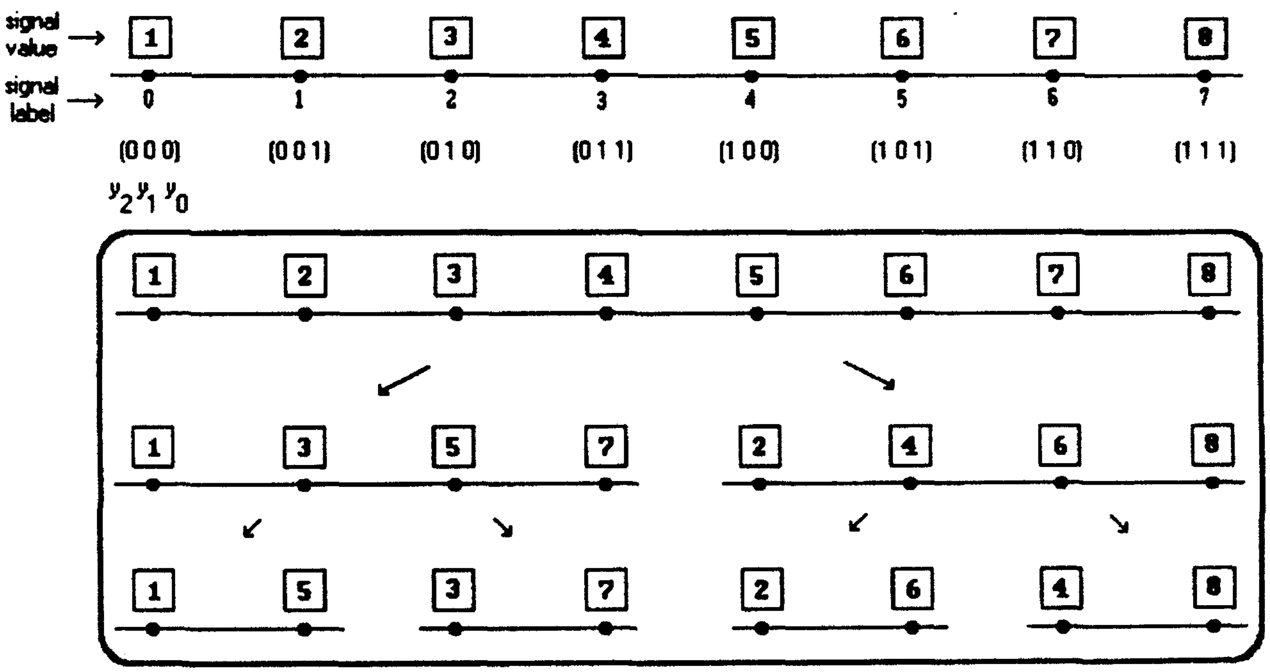


Figure 2.5 8-FSK signal set and set partitioning.

group contains only two signals. The method of division is governed by the principles of set partitioning designed by Ungerboeck. Due to equidistance between any two signals in M-FSK, these partitions are not unique. They are accompanied by rules for signal assignment.

B. SIGNAL ASSIGNMENT PROCEDURE

Coded digits select the signal from the constellation and every interval of T seconds, the encoder transitions to the next state. Uncoded digits select signals corresponding to parallel transitions in the trellis. These parallel transitions do not lend any performance improvement in TCM/M-FSK due to equidistance among all signal pairs. For this reason, they are not considered. The mapper maps the incoming sequences of coded digits to a sequence of M-ary signals. The mapping rule is defined by a function which must be nonlinear in order to achieve a coding gain [Ref. 3].

If k bits are encoded per modulation interval T, there are 2^k possible transitions from each state to a successor state. After selecting a trellis state-transition diagram, one must assign channel signals from the set of 2^{k+1} signals to the transitions such as to achieve maximum free Euclidean distance. We begin with a look at Ungerboeck's three rules for assigning channel signals for amplitude and phase modulations. These same principles may be applied in conjunction with Turgeon's rules to frequency modulation [Ref. 8].

Rules:

U1) All signals should occur with equal frequency and with a fair amount of regularity and symmetry;

U2) Parallel transitions are assigned to signals from the subset with greatest intrasubset distance;

U3) Adjacent transitions (those branches entering or leaving a single state that are not parallel) are assigned to signals from one subset at the final level of set partitioning. [Ref. 6]

These rules are designed to provide codes with good performance. If in addition to the above rules, the following two by Turgeon are applied, the encoder will have minimal complexity, that is, there will be fewer connections in the convolutional encoder [Ref. 7,8].

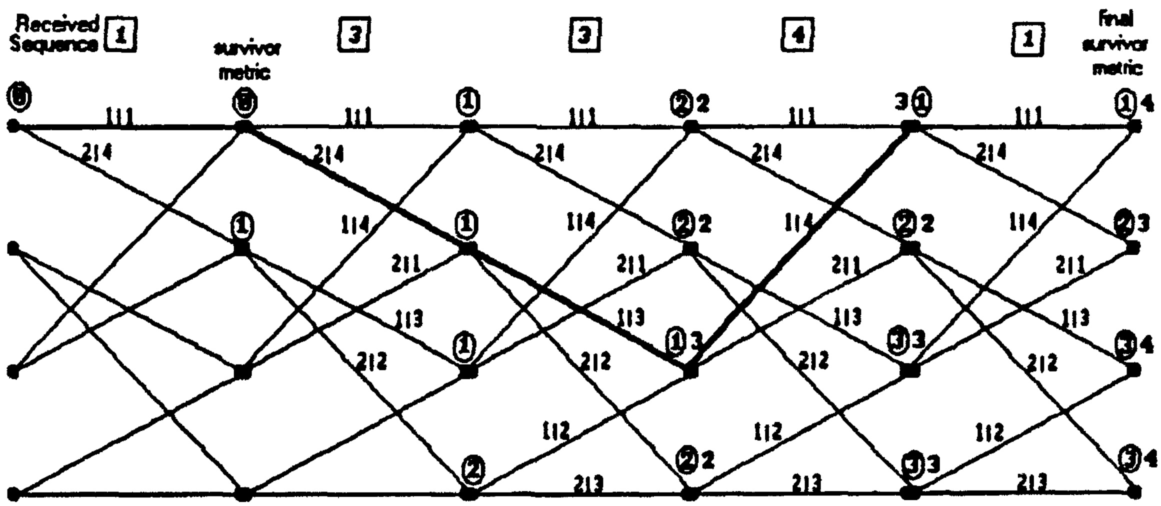
T1) For the signal sequence of state 1: Find the maximum signal value (MSV) at the final level of set partitioning. Choose the MSV and its partner. Assign this pair to state 1. Move up a level and choose the next MSV and its partner. Assign this pair to the next state. Continue until level zero is reached.

T2) In a given dimension (M=4 or M=8), each input bit is associated with a unique signal difference. [Ref. 8]

Regarding rule T2, the signal difference is defined as the difference between signal values attributed to changing one bit in the encoder output from 1 to -1. These differences should be minimal and maintained throughout the trellis.

(Note: While the natural mapper satisfies the guidelines of rule T2, another mapper may be chosen). As minimal complexity doesn't guarantee optimal performance, this and other signal assignment options are examined in the examples of Chapter IV.

The optimum decoder uses the Viterbi algorithm to find the most likely path through the trellis given the observed sequence. Consider the trellis shown in Figure 2.6. Every T seconds, the trellis of Figure 2.3 is repeated thus producing the scheme shown. The object is to find the shortest path through the weighted graph. A detailed description of the Viterbi algorithm is found in Reference 9 and Reference 10.



Binary sequence to be transmitted	:	[0	1	0	0	0]
Uncoded sequence to be transmitted	:	[1	2	1	1	1]
Coded sequence transmitted	:	[1	4	3	4	1]
Received noisy coded sequence	:	[1	3	3	4	1]
Decision coded sequence	:	[1	4	3	4	1]
Viterbi decoder output sequence	:	[1	2	1	1	1]
Binary decoded sequence	:	[0	1	0	0	0]

1 error received

1 error corrected

Figure 2.6 Example Viterbi decoder.

Beginning in the first state, we compute a partial metric for the single path entering each state. For this example, the metric is the Euclidean distance between the two signals, the observed signal and the coded signal on the trellis. This distance is defined to be one for ease of illustration. We store the path (also called the survivor path) and its metric for each state. In the figure, these smallest metrics are circled at each node. Then we compute the partial metric for all paths entering a state by adding the branch metric entering that state to the metric of the connecting survivor at the preceding time unit. When 2^k paths enter a state, the upper path metric is listed at the node first. Then the other paths are listed, with the lowest path metric listed last. Once again, for each state we store the path with the smallest metric and its corresponding metric.

Referring to the trellis shown, we see that the smaller metric from the two branches (upper/lower) entering a node is circled and is the survivor. We decide the rule for a tie is to choose the upper branch. Over time, we see that the path in bold type is the survivor with the smallest metric equal to 1. The Viterbi decoder in this case corrected one error and decided the correct binary sequence was sent.

Although it is not the case of our example, the path chosen may not coincide with the correct path for all time, but will occasionally diverge from it and recombine at a later

time. This is called an error event. The error event length is the number of branches or signals in which the two paths differ.

The error event is caused by the added noise in the transmission medium. To be precise, an error event is formed by "two paths that start in the same state, finish in the same state, and do not simultaneously occupy the same state in between." [Ref. 3] In TCM/M-FSK the minimum length of a possible error event, L, is directly related to the free distance of the code. The free distance is the minimum Euclidean distance of all error events. Both of these parameters are directly involved in the code's performance. The squared Euclidean distance d^2 between any two paths forming an error event of length L is

$$d^2=\sum_{j=1}^{L}|\hat{s}_j-s_j|^2. \qquad (2.1)$$

Here $\mathbf{s}_j$ is the transmitted signal and $\hat{\mathbf{s}}_j$ is the corresponding signal in the diverging branch. Figure 2.7 depicts an error event whose squared euclidean distance is: $d^2 = (a-b)^2 + (c-d)^2 + (e-f)^2$ [Ref. 11]. The distance associated with an error event is of interest because asymptotically for large SNR's, maximizing d_{free} is synonymous with minimizing the average bit error probability.

We find that the normalized free distance of the code sequences depends only on the length of the different paths before recombining due to the equidistance of the orthogonal signals. Consequently, the best rate k/(k+1) binary convolutional codes are those whose diverging paths consist of long error event lengths.

When evaluating the performance of different coding schemes, comparisons are strictly made on the basis of equal data rate and bandwidth. The free Euclidean distance is embedded in a performance measure called coding gain. The coding gain is defined as the difference between the two values of SNR in dB necessary to achieve the same bit error probability in the uncoded and coded systems [Ref. 5].

The *asymptotic coding gain* (ACG) is the ratio of SNR required for coded operation relative to that required for uncoded operation with the same bandwidth efficiency in the limit of large signal to noise ratios. The ACG for TCM/M-FSK with respect to uncoded 16-FSK is defined as:

$$ACG_{dB} = 10 \log [(1/4)(d_{free}^2/2E)] \quad \text{for TCM/4-FSK} \qquad (2.2)$$

$$ACG_{dB} = 10 \log [(1/2)(d_{free}^2/2E)] \quad \text{for TCM/8-FSK} \qquad (2.3)$$

where $d^2_{free}/2E$: normalized squared free Euclidean distance of the coded scheme. This is the smallest of the Euclidean distances between any two coded paths constituting an error event (E: the energy of the coded signal).

A general analytic equation is not available for the free distance, but it may be found by observing the trellis structure and signal assignment. A computer program designed to find the normalized squared free distance given a TCM/M-FSK trellis is found in Reference 1. When the signal set is orthogonal, the distance between any pair of M symbols in the signal constellation is constant. Thus, this is the minimum Euclidean distance between any signal pair. In the computer program this distance is normalized to unity, i.e. d_{ij} is set equal to 1, for all i,j. This is equivalent to finding the minimum error event length, L. Then $d_{free}^2/2E$ is equal to L.

The figure of merit to compare different coding schemes is ACG given by (2.2) and (2.3). The reference uncoded scheme is the one which has the same bandwidth efficiency as the coded scheme. One notes that for *phase shift keying* (PSK) schemes, this means the coded signal space is expanded with reference to the uncoded one. For FSK, however, one reduces the dimension of the signal space when coding to maintain consistent throughput and bandwidth.

Therefore, the bandwidth efficiency R/B is used to find comparable uncoded and coded systems to evaluate coding gains. R is the information rate in bits per second. B is the one-sided bandwidth in Hertz.

For uncoded M-FSK signalling, where $M_u=2^p$ the information rate is $R=p/T_u$ where T_u is the uncoded symbol duration. In

this case the transmission bandwidth is $B=M_u/2T_u$. So, the uncoded scheme's bandwidth efficiency is

$$\left(\frac{R}{B}\right)_u=\frac{2p}{M_u}=\frac{2p}{2^p}. \tag{2.4}$$

For a M-FSK convolutional code with rate $r=k/m$ and $M_c=2^m$, the information rate is likewise $R=k/T_c$, where T_c is the coded symbol duration. For this code, the bandwidth is $B=M_c/2T_c$. Then the coded scheme's bandwidth efficiency is

$$\left(\frac{R}{B}\right)_c=\frac{2k}{M_c}=\frac{2k}{2^m}. \tag{2.5}$$

The ACG ratio is calculated for coded versus uncoded schemes where

$$\left(\frac{R}{B}\right)_u=\left(\frac{R}{B}\right)_c. \tag{2.6}$$

As an example, consider a rate 1/2 TCM/4-FSK code. Note: $k=1$; $m=2$. So, the bandwidth efficiency is $(R/B)_c = 2k/2^m = 1/2$ bits/sec/Hz. Uncoded 16-FSK has $p=4$, so its bandwidth efficiency is $(R/B)_u = 2p/2^p = 1/2$ bits/sec/Hz. Note that rate 2/3 TCM/8-FSK codes have $(R/B)_c = 1/2$ bits/sec/Hz.

III. ENCODER DESIGN

As mentioned in the introduction, the binary stream $\{a_i\}$, $a_i=0,1$ over time is encoded into a sequence of real numbers $\{x_i\}$, $x_i=1,\ldots,M$. Reference 11 contains an in depth discussion on the analytic description of trellis codes. Each channel input x_i depends on the $n=k+v$ most recent bits that enter the encoder. Since each x_i is real, it may be written as a sum of products of the a_i.

$$x(a_1,\ldots,a_n) = c_0 + \sum_{i=1}^{n} c_i a_i + \sum_{\substack{i,j=1 \\ j>i}}^{n} c_{ij} a_i a_j + \ldots + c_{1\ldots n} a_1 a_2 \ldots a_n. \tag{3.1}$$

We note that $x(a_1,\ldots,a_n)$ can take on 2^n values. This is the same as the number of transitions in the trellis. Computationally, it is easier to let the binary data be $\{b_i\}$, $b_i = \pm 1$. The relationship between a_i and b_i is given by the linear transformation $b_i=1-2a_i$, $i=1,2,\ldots,n$. With these antipodal values we can represent the channel input as

$$x(b_1,\ldots,b_n) = d_0 + \sum_{i=1}^{n} d_i b_i + \sum_{\substack{i,j=1 \\ j>i}}^{n} d_{ij} b_i b_j + \sum_{\substack{i,j,h=1 \\ h>j>i}}^{n} d_{ijh} b_i b_j b_h$$

$$+ \ldots + d_{1\ldots n} b_1 b_2 \ldots b_n. \tag{3.2}$$

Equation (3.2) may be put in vector notation. Let $\mathbf{x}$ denote a 2^n element column vector of channel signals assigned in the trellis. Let $\mathbf{d}$ denote the vector of unknown coefficients. Finally, let $\mathbf{B}$ be a 2^n X 2^n matrix where each row represents the 2^n values taken by all the products of the b_i's in (3.2) for each n-tuple $b_1,\ldots,b_n$. Now (3.2) can be written as

$$\mathbf{x} = \mathbf{B}\mathbf{d} \tag{3.3}$$

where

$$x=\begin{bmatrix} x(1,1,\ldots\ldots,1) \\ x(-1,1,\ldots\ldots,1) \\ \vdots \\ x(-1,-1,\ldots,-1) \end{bmatrix}$$

$$\mathbf{B}_i = [\ 1\ b_1\ b_2 \ldots b_n,\ b_1b_2,\ b_2b_3, \ldots, b_1b_2\ldots b_n\ ]$$

and

$$\mathbf{d}^T = [\ d_0\ d_1\ d_2, \ldots, d_{12\ldots n}\ ].$$

As $\mathbf{B}$ is a Hadamard matrix (which is also an orthogonal matrix), then $\mathbf{d}$ is the Hadamard transform of the vector $\mathbf{x}$. We solve for $\mathbf{d}$ by using $\mathbf{B}^{-1} = 1/2^n\ \mathbf{B}^T$:

$$d=\frac{1}{2^n}B^T x. \tag{3.4}$$

So the coefficients are computed by using the signals assigned to all transitions in the trellis diagram and the n bits on which the channel signal depends. Once this relationship between b_i and $\mathbf{x}$ is established, the solution is extended to the logic variables $\{a_i\}$. These new values directly describe the connections in the encoder necessary to generate the desired trellis and signal assignment. The examples in the next chapter will illustrate these concepts.

IV. ENCODER / DECODER DESIGN EXAMPLES

A. INTRODUCTION

This chapter contains five encoder design examples chosen to illustrate different aspects of trellis encoder design. The first three examples are rate 1/2, 4-FSK codes. The first two of these use a simple 4 state code to show the effects of using Ungerboeck's rules and Turgeon's rules respectively. It is determined that they do not provide any coding gain relative to uncoded 16-FSK. The third example uses a 64 state code and Ungerboeck's rules to show that the squared free Euclidean distance (thus performance) increases with increasing the number of shift registers v, as compared with the earlier examples. In this example, the asymptotic coding gain is 2.43 dB. All rate 1/2 codes are checked for decodability with a Viterbi decoder simulation program.

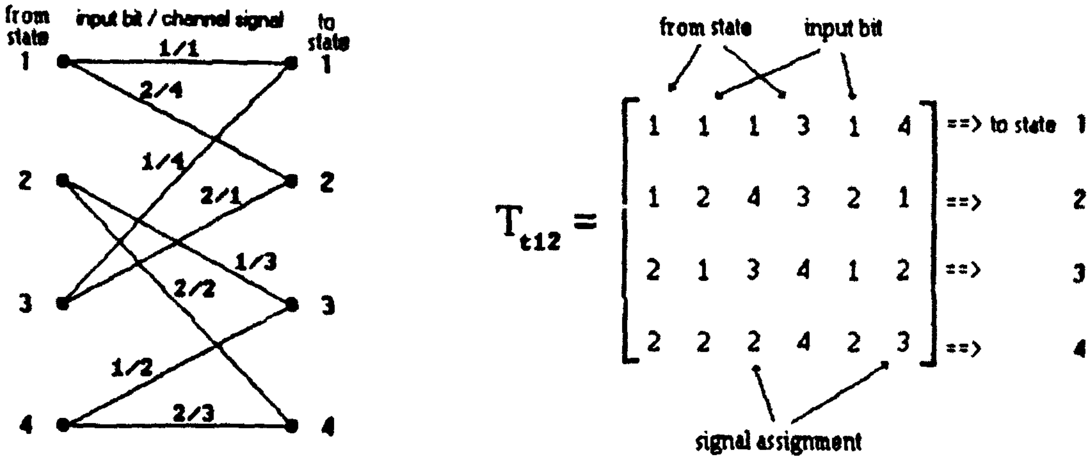
The last two examples are rate 2/3, 8-FSK codes. The first of these uses an 8 state code and Turgeon's rules to derive the minimal complexity encoder. It is determined that this code does not provide any coding gain due to a small error event length. The final example uses a 32 state code and Turgeon's rules to acieve an asymptotic coding gain of 1.76 dB.

B. DESIGN EXAMPLE 1: 4-FSK, RATE 1/2, 4 STATE

For this example, all details are explained to ensure understanding of the methods previously described. While the program is designed to do all the computations, the logical steps are provided here in parallel for verification purposes.

This example is provided to illustrate the rules of Ungerboeck in a simple code. In this scheme, there is one input data bit (k=1) and two memory elements (v=2). Thus the channel signals depend on the three most recent bits (n=k+v=3). Figure 2.1 describes the relationship between the data and memory bits. However, in this case, the encoder connections are unknown. The signal set and its partitioning are shown in Figure 2.4.

To find the underlying convolutional encoder we start with the trellis in Figure 4.1. The information in the trellis is equivalently described by the matrix $\mathbf{T}_{t12}$ [Ref. 12]. This figure also contains the program input and a sample run. Here, the program input is described. The program is designed to intake the trellis in the form of a matrix $\mathbf{T}$ ($\mathbf{T}_{t12}$ to distinguish it from other examples). For simplicity, we will only look at the first three columns of $\mathbf{T}_{t12}$. Each row number r corresponds to information going to state number r. For example, row 1: transition goes from state 1 to state 1 with input bit 1 and channel signal assignment 1.



Signal assignment trellis for rate 1/2, 4-FSK, 4 state code with corresponding T matrix

```
DESIGN EXAMPLE 1:  STARTING WITH TRELLIS, rate 1/2, v=2
£ tt        % Running tt.m puts matrix Tt12 into the workspace.
  Tt12 =    1     1     1     3     1     4
            1     2     4     3     2     1
            2     1     3     4     1     2
            2     2     2     4     2     3
£ tcmmain
Program USER INPUT:
        Starting with Trellis?  (y/n)                : y
        Choose Code Rate:  1) 1/2, or 2) 2/3; rate: 1
        Enter the Trellis matrix:               T =   Tt12
Program OUTPUT:
        The number of input bits is                  : k = 1
        The number of memory bits is                 : v = 2
The analytic description b coefficients are :  -0.5   -1.0
The b's to connect are                      :  0   1   3
                                               1   2   3
The logic variables (a's) to connect are    : y0 =   1   3
                                              y1 =   1   2   3
dfree^2 (normalized) =  3
Your encoder is decodable.
```

Figure 4.1 Design Example 1: Trellis, matrix T, and program run

Row 2: transition goes from state 1 to state 2 with input bit 2 and signal assignment 4. For each of the 2^k transitions from a node in the trellis, there is a set of three columns. This explains why there are six (2^k*3) total columns in the matrix. Also note that there are four (2^v) rows (or states) in the matrix $\mathbf{T}_{\mathrm{u}12}$. The program output is the encoder connections needed to create the given trellis assuming the natural mapper is used. The program is restricted to values of $n<10$, although the procedure is valid for $n\geq10$.

When approaching the problem theoretically, one must first determine how the output bits relate to the channel signal. Utilizing the natural mapper and signal difference information depicted in Table 4.1, the relationship between the channel signal and the output bits is

$$x = 2.5 - z_1 - 0.5z_0. \tag{4.1}$$

Table 4.1

Signal Mapping and Signal Differences for 4-FSK Signal Set						
MSB y_1	LSB y_0	MSB z_1	LSB z_0	Signal label	Signal Value : Channel Signal x	Signal Differences
0	0	1	1	0	1	z_1: (1-3) $= \delta 1$
0	1	1	-1	1	2	z_0: (1-2) $= \delta 0$
1	0	-1	1	2	3	d_{z1}: $= (1/2)\delta 1 = -1.0$
1	1	-1	-1	3	4	d_{z0}: $= (1/2)\delta 0 = -0.5$

Solving for **d** in (3.4), the coefficients are

$d_0 = 2.5$

$d_{13} = -0.5$

$d_{123} = -1.0.$

So, following (3.2) the analytic description of the trellis code is

$$x = 2.5 - 0.5 b_1 b_3 - b_1 b_2 b_3 \tag{4.2}$$

Comparing (4.1) and (4.2), it is clear that

$z_1 = b_1 b_2 b_3$

$z_0 = b_1 b_3.$

The analytic description transmitter may be implemented as in Figure 4.2.

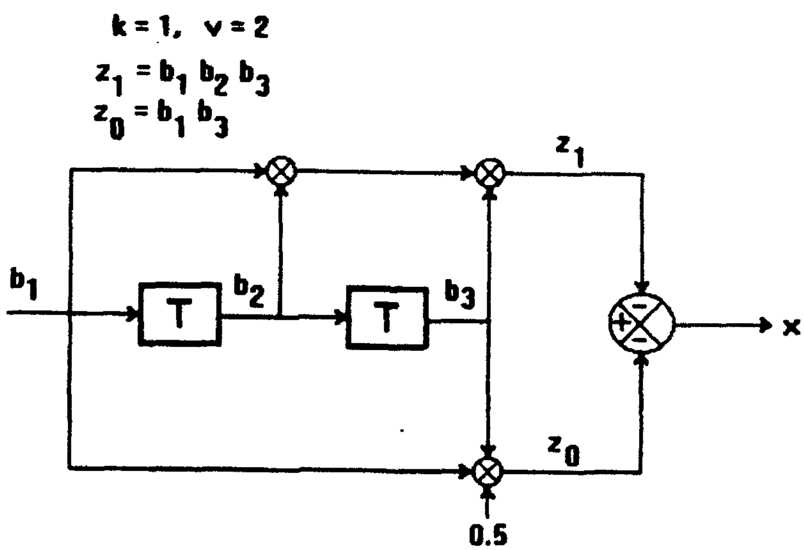


Figure 4.2 TCM encoder analytic description for rate 1/2, 4-FSK, 4 state code.

Due to the relationship $a_i = (1-b_i)/2$, $i=1,\ldots,n$, the corresponding relations for the logic variables, a_i $\{0,1\}$ are

$$y_0 = a_1 \oplus a_3$$

$$y_1 = a_1 \oplus a_2 \oplus a_3,$$ where $\oplus$ is the modulo 2 sum.

Using these connections, the convolutional encoder is given in Figure 4.3.

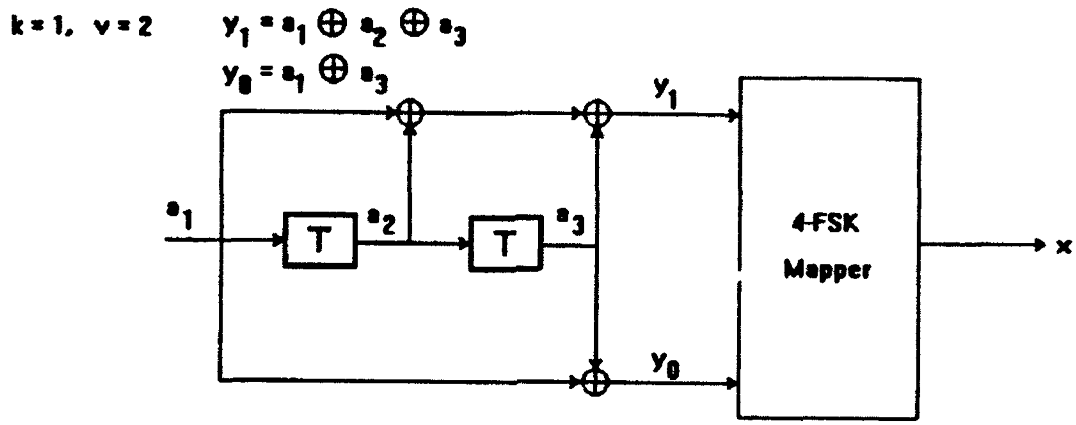


Figure 4.3 TCM encoder for rate 1/2, 4-FSK, 4 state code.

From Figure 2.6 the minimum error event length is 3. The normalized squared free Euclidean distance attributed to this error event is $d_{free}^2/2E = 3$. The actual d_{free}^2 is $(\sqrt{2E}^2 + \sqrt{2E}^2 + \sqrt{2E}^2) = 3(2E)$. The asymptotic coding gain of this scheme using (2.2) relative to uncoded 16-FSK modulation which has the same bandwidth efficiency is -1.25 dB. This example implies that a greater minimum error event length must be achieved by a code in order to get a coding gain.

The program tests the encoder in a 100 bit noiseless simulation to ensure the signal assignment does not contribute

to a catastrophic code. A catastrophic trellis code is one where "more and more paths appear which have a squared distance equal to or slightly larger than the squared free distance [Ref. 13]." When this happens, the free distance alone is not a good indicator of the coding gain, as the probability of selecting one of these close incorrect paths increases.

Since most codes are described by their generator polynomials, one may also start the program given the rate 1/2 encoder connections. This option in the program outputs the trellis **T**, $d_{free}^2/2E$, and checks for decodability. Figure 4.4 contains a sample run for input $y_0 = a_1 \oplus a_3$ and $y_1 = a_1 \oplus a_2 \oplus a_3$. The trellis produced is $\mathbf{T} = \mathbf{T}_{t12}$.

```
DESIGN EXAMPLE 1: STARTING WITH ENCODER CONNECTIONS, rate 1/2,
£ tcmmain                                                    v=2
Program USER INPUT:
        Starting with Trellis?                  (y/n) : n
        Starting with Encoder connections?      (y/n) : y

        Enter the number of input bits :       k = 1
        Enter the number of memory bits:       v = 2
        Enter LSB encoder connections  :       y0= [1 3]
        Enter MSB encoder connections  :       y1= [1 2 3]
Program OUTPUT:
        The trellis signal matrix is: T =      1  1  1  3  1  4
                                               1  2  4  3  2  1
                                               2  1  3  4  1  2
                                               2  2  2  4  2  3
dfree^2 (normalized) = 3
Your encoder is decodable.
```

Figure 4.4 Design Example 1: Starting with encoder connections

In the simulation program, the Viterbi decoder decision delay or path history is defined to be 6v, or 12 in this case [Ref. 6]. As the encoder is decodable (there were no errors in the decision), one can run the simulation programs with an input noise level.

A sample run for 17 bits is found in Figure 4.5. As the decision delay is 12 bits, the decoder will decode the first 5 bits. The user inputs the symbol energy. Here it is chosen equal to 1. The standard deviation of the noise is chosen to be 0.3.

The simulation program output for the transmitter portion consists of a random message to be transmitted, the encoder output, the signals depicted by the natural mapper, and the sequence of frequencies to be transmitted. Each row of the matrix **M** is a four dimensional vector as in (1.3) that represents the signal to be transmitted. Noise is added to this matrix element by element.

In the receiver, the decoder needs information about the trellis and the current decision statistics vector for each signal in the received sequence. **U** is a matrix of metrics. Each row of matrix **U** is a Euclidean distance measurement between the received signal (in the same row of **M**) and each of the four possible signals in 4-ary modulation. The distance between the signal in the first row of **M** and $\mathbf{s}_1$ in (1.4) is placed in the first column of **U** (row 1).

```
DESIGN EXAMPLE 1: SIMULATION FOR Tt12 TRELLIS, rate 1/2, v=2

The trellis signal matrix is: Tt12  =  1   1   1   3   1   4
                                       1   2   4   3   2   1
                                       2   1   3   4   1   2
                                       2   2   2   4   2   3

Program USER INPUT:
Run simulation?   (y/n)                 :  y

     THIS IS THE VITERBI DECODER FOR RATE 1/2 TCM-FSK
          There is a 12 bit delay in the decoder.
          The message length must be longer than this.

How many bits are in the message?       :    nb = 17
Enter the symbol energy, eg. Es=1.      :    Es = 1

Es/No=1/2*sigma^2 where sigma is the AWGN standard deviation.
Enter the value of sigma, eg. sigma=0.1 : sigma = 0.3

Program OUTPUT:
The random message to be transmitted is:
      rnd_o = 1 0 1 1 1 0 0 0 1 0 1 1 1 1 1 1 1

The random message with shift register memories set = 0.
  rnd_o = 0 0 1 0 1 1 1 0 0 0 1 0 1 1 1 1 1 1 1

The output of the encoder is LSB,MSB:
     encout = 1 0 0 1 0 1 1 0 1 0 0 1 0 0 0 0 0
              1 1 0 0 1 0 1 0 1 1 0 0 1 1 1 1 1

The symbol sequence equivalent is:
          m = 3 2 0 1 2 1 3 0 3 2 0 1 2 2 2 2 2

The sequence of frequencies is:
          f = 4 3 1 2 3 2 4 1 4 3 1 2 3 3 3 3 3

Symbol matrix, M, before noise is added:
    M =  0     0     0     1
         0     0     1     0
         1     0     0     0
         0     1     0     0
         0     0     1     0
         0     1     0     0
         0     0     0     1
         1     0     0     0
         0     0     0     1
         0     0     1     0
         1     0     0     0
         0     1     0     0
         0     0     1     0
         0     0     1     0
         0     0     1     0
         0     0     1     0
         0     0     1     0
```

Figure 4.5a Design Example 1: Simulation

```
Symbol matrix, M, with AWGN:
M =
-0.1184+0.4552i -0.0199-0.5514i  0.1556-0.4551i  0.6980-0.2017i
-0.2345-0.7747i  0.6004-0.5735i  0.8901-0.0381i  0.0340-0.4548i
 0.6961-0.2910i -0.3430-0.0901i  0.3336-0.0611i -0.7341+0.1505i
 0.0751+0.0844i  1.2597+0.6466i  0.1481-0.0457i  0.4314-0.2576i
 0.1399+0.2330i -0.4667-0.1820i  0.9724-0.7803i -0.0074-0.3830i
-0.0659-0.0720i  0.8214+0.1602i -0.0248-0.2700i -0.1311-0.0037i
 0.6524+0.1441i -0.2856+0.0827i  0.0627+0.1563i  1.1764-0.5669i
 0.4891-0.1650i  0.1804-0.0994i  0.1345-0.1347i -0.0899+0.4027i
-0.1398+0.1019i -0.3223+0.1779i  0.4548-0.1045i  1.6042+0.0980i
 0.3267-0.1936i -0.2230-0.4833i  0.8274-0.5360i -0.2870-0.0313i
 0.7524+0.6328i  0.1125+0.3795i -0.1160+0.2902i -0.2392+0.0294i
 0.1213-0.0052i  1.5880+0.0290i -0.0997-0.1319i  0.3642+0.3687i
-0.2824+0.0516i -0.5079+0.0749i  0.7809-0.1674i -0.0023+0.2476i
 0.2775+0.0527i -0.5075-0.5585i  0.7902-0.0749i  0.4605+0.0774i
-0.1406+0.5448i -0.0533+0.3427i  0.8832+0.2856i -0.0348-0.0753i
-0.1788-0.0397i  0.1811-0.1408i  1.4757+0.1984i  0.1417+0.0007i
-0.2578+0.0781i  0.2426+0.1040i  1.4840+0.0905i -0.1372+0.0797i

The Decision statistics vectors sent to the Viterbi decoder is
    U =  1.5880      1.5247      1.4049      0.9429
         1.9533      1.4647      1.2515      1.8106
         0.9896      1.7486      1.3055      1.9595
         1.7731      0.8802      1.7314      1.5593
         1.6571      1.9898      1.0398      1.7438
         1.3829      0.3712      1.3528      1.4293
         1.4013      1.9595      1.7729      0.9569
         0.7329      1.0745      1.1164      1.3020
         2.0606      2.1473      1.7483      0.8706
         1.3525      1.7114      0.9099      1.7484
         0.8798      1.4331      1.5846      1.6605
         1.8949      0.8106      2.0081      1.7620
         1.6155      1.7495      0.6952      1.4316
         1.3937      1.8741      0.9575      1.2555
         1.6082      1.5530      0.7341      1.5410
         1.9186      1.7208      0.6099      1.7435
         1.9732      1.7008      0.6402      1.9111

DECODED BIT MSG SEQUENCE (Note: -1 represents delay.) :
s_hat   = -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1  0  1  1  1

INPUT BIT MSG SEQUENCE DELAYED BY 6*v BITS:
s_delay = -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1  0  1  1  1

NUMBER of BIT ERRORS in a 17 bit message is:  n = 0
```

Figure 4.5b Design Example 1: Simulation (continued)

Similarly, the first row of **M** is compared to $\mathbf{s}_2$ and placed in column 2 of **U** (row 1). When complete, **U** is a M X nb (4 X 17) matrix, where nb is the number of bits in the message.

The simulation program computes the decoded message sequence and compares it to a delayed version of the irput sequence. The number of errors in the sample run is zero.

C. DESIGN EXAMPLE 2: 4-FSK, RATE 1/2, 4 STATE

This example is provided to illustrate the application of Turgeon's rules. If correctly applied, the resulting encoder design should be of minimal complexity. Since the basic scheme is the same as was described in Design 4.1, one can compare the implementation complexity of the two designs. Here, only the program input and output are described. In this scheme, there is one input data bit (k=1) and two memory elements (v=2). Thus the channel signals depend on the three most recent bits (n=k+v=3). Once again Figure 2.1 describes the relationship between the data and memory bits. To find the underlying convolutional encoder we start with the trellis and its matrix $\mathbf{T}_{j12}$ in Figure 4.6. The signal set and its partitioning are also shown. Figure 4.7 contains the program run.

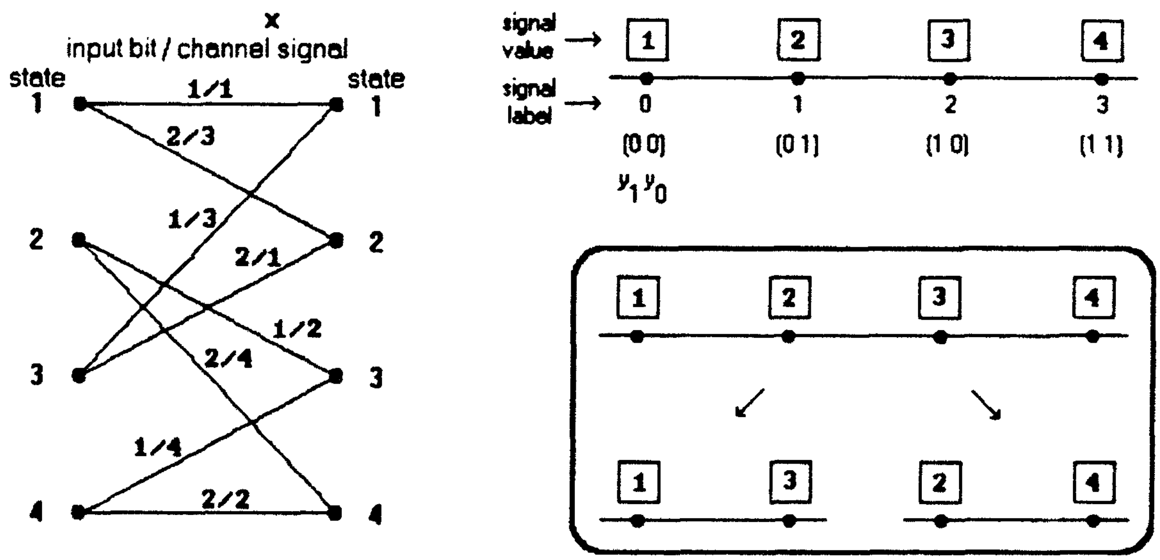


Figure 4.6 4-FSK trellis, signal set and set partitioning.

```
DESIGN EXAMPLE 2: STARTING WITH TRELLIS, rate 1/2, v=2

£ tt
  % Running tt.m puts matrix Tj12 into the workspace.
  Tj12 =
     1      1      1      3      1      3
     1      2      3      3      2      1
     2      1      2      4      1      4
     2      2      4      4      2      2

£ tcmmain
Program USER INPUT:
    Starting with Trellis?  (y/n)                        :  y
    Choose Code Rate:  1)  1/2,   or   2)  2/3;   rate:  1
    Enter the Trellis matrix:                      T =    Tj12

Program OUTPUT:
    The number of input bits is : k = 1
    The number of memory bits is: v = 2

    The analytic description b coefficients are:   -0.5   -1.0
    The bs to connect are:       0      0      2
                                 0      1      3

    The logic variables (a's) to connect are:  y0 =  2
                                               y1 =  1      3
    dfree^2 (normalized) =  3
    Your encoder is decodable.
```

```
DESIGN EXAMPLE 2: STARTING WITH ENCODER CONNECTIONS, rate 1/2,
                                                             v=2
£ tcmmain

Program USER INPUT:
  Starting with Trellis?  (y/n)               : n
  Starting with Encoder connections?  (y/n)  : y

  Enter the number of input bits:   k = 1
  Enter the number of memory bits:  v = 2
  Enter LSB encoder connections    y0 = [2]
  Enter MSB encoder connections    y1 = [1 3]

Program OUTPUT:
  The trellis signal matrix is:  T =  1    1    1    3    1    3
                                      1    2    3    3    2    1
                                      2    1    2    4    1    4
                                      2    2    4    4    2    2
dfree^2 (normalized) =  3
Your encoder is decodable.
```

Figure 4.7 Design Example 2: Trellis, matrix T, and program run

The program run shows the program input, matrix $\mathbf{T}=\mathbf{T}_{j12}$, and the encoder connections needed to create the given trellis assuming the natural mapper in Table 4.1 is used. The analytic code description is

$$x = 2.5 - 0.5b_2 - b_1b_3$$

Following the procedure of Design Example 1,

$z_1 = b_1b_3$

$z_0 = b_2.$

The corresponding relations for the logic variables, a_i $\{0,1\}$ are

$y_0 = a_2$

$y_1 = a_1 \oplus a_3$, where $\oplus$ is the modulo 2 sum.

Using these connections, the convolutional encoder is given in Figure 4.8. The complexity is minimal as each shift register is only contained once in the encoder output equations.

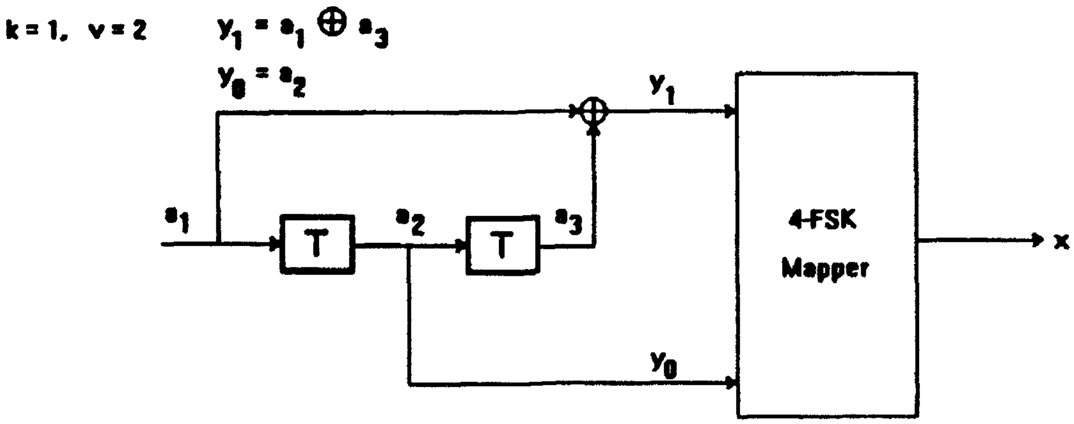


Figure 4.8 Minimal complexity TCM encoder for rate 1/2, 4-FSK, 4 state code.

Once again Figure 2.6 depicts the minimum error event length to be 3. The normalized squared free Euclidean distance attributed to this error event is $d_{free}^2/2E = 3$. The asymptotic coding gain of this scheme using (2.2) relative to uncoded 16-FSK modulation which has the same bandwidth efficiency is -1.25 dB.

The encoder is determined decodable by the 100 bit noiseless simulation program.

D. DESIGN EXAMPLE 3: 4-FSK, RATE 1/2, 64 STATE

In this scheme, there is one input data bit (k=1) and six memory elements (v=6). Thus the channel signals depend on the seven most recent bits (n=k+v=7). Here Figure 2.1 describes the relationship between the data and memory bits. The signal set and its partitioning are shown in Figure 2.4. The natural mapper in Table 4.1 is assumed. To find the underlying convolutional encoder we start with the trellis described by matrix $\mathbf{T}_{t16}$ in Figure 4.9. The program output in Figure 4.10 contains the encoder connections needed to create the given trellis. The analytic code description is

$$x = -0.5b_1b_3b_4b_5b_6b_7 - b_1b_2b_4b_5b_7.$$

Following the procedure of Design Example 1,

$$z_1 = b_1b_2b_4b_5b_7$$

$$z_0 = b_1b_3b_4b_5b_6b_7.$$

The corresponding relations for the logic variables, a_i $\{0,1\}$ are

$$y_0 = a_1 \oplus a_3 \oplus a_4 \oplus a_5 \oplus a_6 \oplus a_7$$

$$y_1 = a_1 \oplus a_2 \oplus a_4 \oplus a_5 \oplus a_7.$$

Using these connections, the convolutional encoder is given in Figure 4.11.

The minimum error event length is 7. The normalized squared free Euclidean distance attributed to this error event is $d_{free}^2/2E = 7$. The actual d_{free}^2 is 7(2E).

```
DESIGN EXAMPLE 3: STARTING WITH TRELLIS, rate 1/2, v=6
% tt  % Running tt.m puts matrix Tt16 into the workspace.
Tt16=1     1     1    33     1     4
     1     2     4    33     2     1
     2     1     3    34     1     2
     2     2     2    34     2     3
     3     1     2    35     1     3
     3     2     3    35     2     2
     4     1     4    36     1     1
     4     2     1    36     2     4
     5     1     4    37     1     1
     5     2     1    37     2     4
     6     1     2    38     1     3
     6     2     3    38     2     2
     7     1     3    39     1     2
     7     2     2    39     2     3
     8     1     1    40     1     4
     8     2     4    40     2     1
     9     1     4    41     1     1
     9     2     1    41     2     4
    10     1     2    42     1     3
    10     2     3    42     2     2
    11     1     3    43     1     2
    11     2     2    43     2     3
    12     1     1    44     1     4
    12     2     4    44     2     1
    13     1     1    45     1     4
    13     2     4    45     2     1
    14     1     3    46     1     2
    14     2     2    46     2     3
    15     1     2    47     1     3
    15     2     3    47     2     2
    16     1     4    48     1     1
    16     2     1    48     2     4
    17     1     2    49     1     3
    17     2     3    49     2     2
    18     1     4    50     1     1
    18     2     1    50     2     4
    19     1     1    51     1     4
    19     2     4    51     2     1
    20     1     3    52     1     2
    20     2     2    52     2     3
    21     1     3    53     1     2
    21     2     2    53     2     3
    22     1     1    54     1     4
    22     2     4    54     2     1
    23     1     4    55     1     1
    23     2     1    55     2     4
    24     1     2    56     1     3
    24     2     3    56     2     2
    25     1     3    57     1     2
    25     2     2    57     2     3
    26     1     1    58     1     4
    26     2     4    58     2     1
    27     1     4    59     1     1
    27     2     1    59     2     4
    28     1     2    60     1     3

continued:
28     2     3    60     2     2
29     1     2    61     1     3
29     2     3    61     2     2
30     1     4    62     1     1
30     2     1    62     2     4
31     1     1    63     1     4
31     2     4    63     2     1
32     1     3    64     1     2
32     2     2    64     2     3
```

Figure 4.9 Design Example 3: Trellis matrix, T

```
DESIGN EXAMPLE 3: STARTING WITH TRELLIS, rate 1/2, v=6

£ tcmmain
Program USER INPUT:
 Starting with Trellis?  (y/n)                          : y
 Choose Code Rate:  1)  1/2,    or    2)  2/3;    rate: 1
 Enter the Trellis matrix:                        T =   Tt16

Program OUTPUT:
 The number of input bits is : k = 1
 The number of memory bits is: v = 6

 The analytic description b coefficients are:   -1.0    -0.5
 The b's to connect are:  0      0      1      2      4      5      7
                          0      1      3      4      5      6      7

 The logic variables to connect are: y0 =  1  3  4  5  6  7
                                     y1 =  1  2  4  5  7

 dfree (normalized) = 7
 Your encoder is decodable.
```

```
DESIGN EXAMPLE 3: SIMULATION FOR Tt16 TRELLIS, rate 1/2,
                                                      v=6

Program USER INPUT:
Run simulation?   (y/n)                     :  y

     THIS IS THE VITERBI DECODER FOR RATE 1/2 TCM-FSK
          There is a 36 bit delay in the decoder.
          The message length must be longer than this.

How many bits are in the message?          :    nb = 1000
Enter the symbol energy, eg. Es=1.         :    Es = 1
  Es/No=1/2*sigma^2 where sigma is the AWGN standard deviation.
Enter the value of sigma, eg. sigma=0.1 : sigma = 0.3

NUMBER of BIT ERRORS in a 1000 bit message is:   n =  0
```

Figure 4.10 Design Example 3: Program run and simulation

$k = 1, \quad v = 6 \qquad y_1 = a_1 \oplus a_2 \oplus a_4 \oplus a_5 \oplus a_7$

$$y_0 = a_1 \oplus a_3 \oplus a_4 \oplus a_5 \oplus a_6 \oplus a_7$$

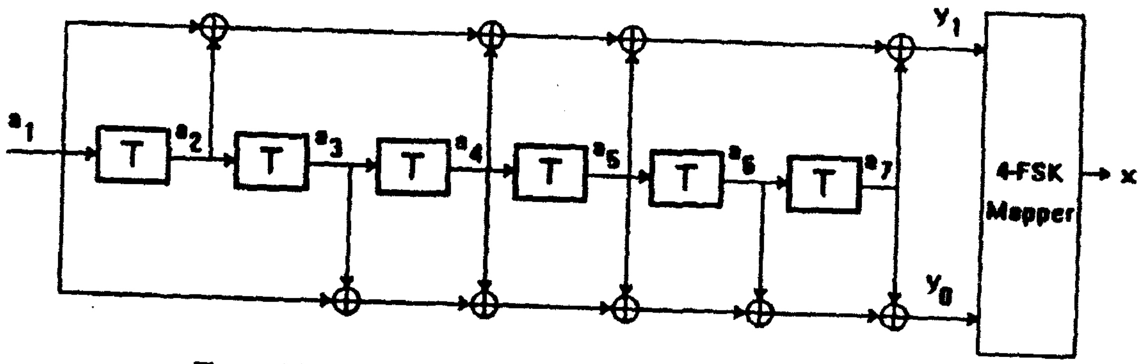


Figure 4.11 TCM encoder for rate 1/2, 4-FSK, 64 state code.

The asymptotic coding gain of this scheme using (2.2) relative to uncoded 16-FSK modulation which has the same bandwidth efficiency is

2.43 dB.

The program tests the encoder in a 100 bit noiseless simulation and asserts that the encoder is decodable. The second portion of Figure 4.10 shows that the 1000 bit simulation results in no errors.

E. DESIGN EXAMPLE 4: 8-FSK, RATE 2/3, 8 STATE

In this scheme, there are two input data bits (k=2) and three memory elements (v=3). Thus the channel signals depend on the five most recent bits (n=k+v=5). Let Figure 4.12 depict the relationship between the input and state variables.

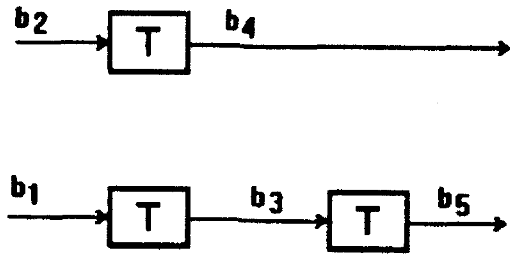


Figure 4.12 Inputs and state variables for rate 2/3 code.

The signal set is shown in Figure 2.5. The natural mapper in Table 4.2 is assumed.

Table 4.2

Signal Mapping and Signal Differences for 8-FSK Signal Set								Signal Differences
MSB y_2	y_1	LSB y_0	MSB z_2	z_1	LSB z_0	Signal label	Signal Value : Channel Signal x	
0	0	0	1	1	1	0	1	
0	0	1	1	1	-1	1	2	z_2: (1-5) = $\delta 2$
0	1	0	1	-1	1	2	3	z_1: (1-3) = $\delta 1$
0	1	1	1	-1	-1	3	4	z_0: (1-2) = $\delta 0$
1	0	0	-1	1	1	4	5	d_{z2}: = (1/2) $\delta 2$ = -2.0
1	0	1	-1	1	-1	5	6	d_{z1}: = (1/2) $\delta 1$ = -1.0
1	1	0	-1	-1	1	6	7	d_{z0}: = (1/2) $\delta 0$ = -0.5
1	1	1	-1	-1	-1	7	8	

To find the underlying convolutional encoder we start with the trellis in Figure 4.13 described by matrix $\mathbf{T}_{j23}$ in Figure 4.14.

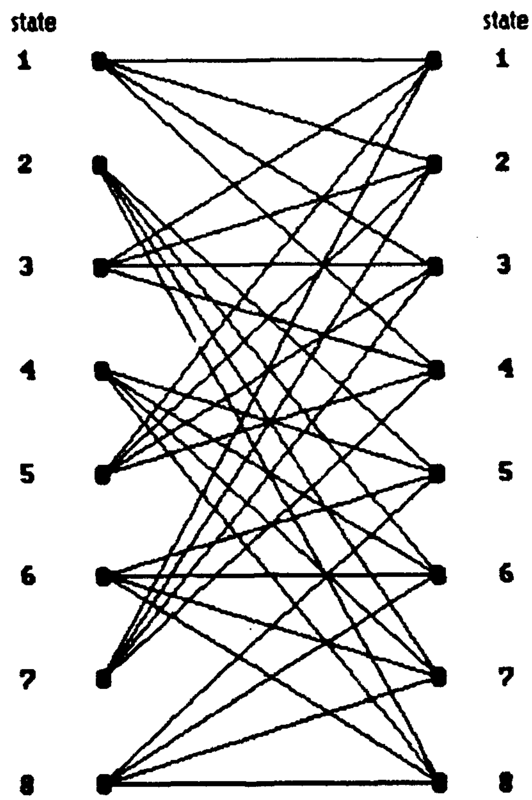


Figure 4.13 Trellis for rate 2/3, 8-FSK, 8 state code.

From the program output, the analytic code description is:

$$x = 4.5 - 0.5b_3 - b_2b_5 - 2b_1b_4$$

Following the procedure of Design Example 1,

$$z_0 = b_3$$

$$z_1 = b_2b_5$$

$$z_2 = b_1b_4.$$

```
DESIGN EXAMPLE 4: STARTING WITH TRELLIS, rate 2/3, v=3

£ tt      % Running tt.m puts matrix Tj23 into the workspace.

Tj23 =
     1    1    1    3    1    5    5    1    3    7    1    7
     1    2    5    3    2    1    5    2    7    7    2    3
     1    3    3    3    3    7    5    3    1    7    3    5
     1    4    7    3    4    3    5    4    5    7    4    1
     2    1    2    4    1    6    6    1    4    8    1    8
     2    2    6    4    2    2    6    2    8    8    2    4
     2    3    4    4    3    8    6    3    2    8    3    6
     2    4    8    4    4    4    6    4    6    8    4    2

£ tcmmain

Program USER INPUT:
 Starting with Trellis?  (y/n)                          : y
 Choose Code Rate:  1)  1/2,    or    2)  2/3;    rate: 2
 Enter the Trellis matrix:                        T =   Tj23

Program OUTPUT:
 The number of input bits is :  k = 2
 The number of memory bits is:  v = 3

 The analytic description b coefficients are:  -0.5  -2.0  -1.0
 The bs to connect are:         0     0     0     0     3
                                0     0     0     1     4
                                0     0     0     2     5

 The logic variables to connect are:       y0 =   3
                                           y1 =   2     5
                                           y2 =   1     4

 dfree^2 (normalized) = 2
```

Figure 4.14 Design Example 4: Starting with trellis

The corresponding relations for the logic variables, a_i {0,1} are

$$y_0 = a_3$$

$$y_1 = a_2 \oplus a_5$$

$$y_2 = a_1 \oplus a_4.$$

Using these connections, the convolutional encoder is given in Figure 4.15. Note the minimal complexity of the encoder.

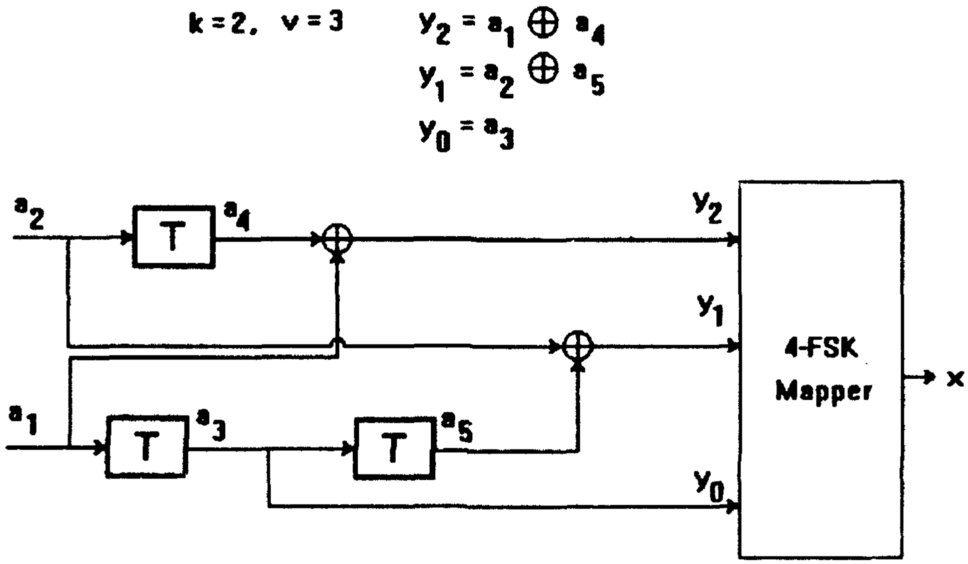


Figure 4.15 Minimal complexity TCM encoder for rate 2/3, 8-FSK, 8 state code.

The minimum error event length is 2. The normalized squared free Euclidean distance attributed to this errcr event is $d_{free}^2/2E = 2$. The actual d_{free}^2 is 2(2E). The asymptotic coding gain of this scheme using (2.3) relative to uncoded 16-FSK modulation which has the same bandwidth efficiency is 0 dB.

F. DESIGN EXAMPLE 5: 8-FSK, RATE 2/3, 32 STATE

In this scheme, there are two input data bits (k=2) and five memory elements (v=5). Thus the channel signals depend on the seven most recent bits (n=k+v=7). The signal set is shown in Figure 2.5. The natural mapper in Table 4.2 is assumed. To find the underlying convolutional encoder we start with the trellis described by matrix $\mathbf{T}_{j25}$ in Figure 4.16. From the program output, the analytic code description is:

$$x = 4.5 - 0.5b_3b_6 - b_2b_5 - 2b_1b_4b_7$$

Following the procedure of Design Example 1,

$z_0 = b_3b_6$

$z_1 = b_2b_5$

$z_2 = b_1b_4b_7.$

The corresponding relations for the logic variables, a_i $\{0,1\}$ are

$y_0 = a_3 \oplus a_6$

$y_1 = a_2 \oplus a_5$

$y_2 = a_1 \oplus a_4 \oplus a_7.$

Using these connections, the convolutional encoder is given in Figure 4.17. Note the minimal complexity of the encoder.

The minimum error event length is 3. The normalized squared free Euclidean distance attributed to this error event

DESIGN EXAMPLE 5: STARTING WITH TRELLIS, rate 2/3, v=5

```
» tt     % Running tt.m puts matrix Tt25 into the workspace.

 Tt25 =
     1     1     1     9     1     2    17     1     5    25     1     6
     1     2     5     9     2     6    17     2     1    25     2     2
     1     3     3     9     3     4    17     3     7    25     3     8
     1     4     7     9     4     8    17     4     3    25     4     4
     2     1     2    10     1     1    18     1     6    26     1     5
     2     2     6    10     2     5    18     2     2    26     2     1
     2     3     4    10     3     3    18     3     8    26     3     7
     2     4     8    10     4     7    18     4     4    26     4     3
     3     1     5    11     1     6    19     1     1    27     1     2
     3     2     1    11     2     2    19     2     5    27     2     6
     3     3     7    11     3     8    19     3     3    27     3     4
     3     4     3    11     4     4    19     4     7    27     4     8
     4     1     6    12     1     5    20     1     2    28     1     1
     4     2     2    12     2     1    20     2     6    28     2     5
     4     3     8    12     3     7    20     3     4    28     3     3
     4     4     4    12     4     3    20     4     8    28     4     7
     5     1     3    13     1     4    21     1     7    29     1     8
     5     2     7    13     2     8    21     2     3    29     2     4
     5     3     1    13     3     2    21     3     5    29     3     6
     5     4     5    13     4     6    21     4     1    29     4     2
     6     1     4    14     1     3    22     1     8    30     1     7
     6     2     8    14     2     7    22     2     4    30     2     3
     6     3     2    14     3     1    22     3     6    30     3     5
     6     4     6    14     4     5    22     4     2    30     4     1
     7     1     7    15     1     8    23     1     3    31     1     4
     7     2     3    15     2     4    23     2     7    31     2     8
     7     3     5    15     3     6    23     3     1    31     3     2
     7     4     1    15     4     2    23     4     5    31     4     6
     8     1     8    16     1     7    24     1     4    32     1     3
     8     2     4    16     2     3    24     2     8    32     2     7
     8     3     6    16     3     5    24     3     2    32     3     1
     8     4     2    16     4     1    24     4     6    32     4     5

» tcmmain
Program USER INPUT:
 Starting with Trellis?  (y/n)                      : y
 Choose Code Rate:  1)  1/2,    or    2)  2/3;   rate: 2
 Enter the Trellis matrix:                       T =   Tt25

Program OUTPUT:
 The number of input bits is :  k = 2
 The number of memory bits is:  v = 5

 The analytic description b coefficients are: -1.0  -0.5  -2.0
 The b's to connect are:   0    0    0    0    0    2    5
                           0    0    0    0    0    3    6
                           0    0    0    0    1    4    7

The logic variables to connect are:   y0 =  3     6
                                      y1 =  2     5
                                      y2 =  1     4     7
dfree^2 (normalized) = 3
```

Figure 4.16 Design Example 5: Starting with trellis

$$y_2 = a_1 \oplus a_4 \oplus a_7$$

$k = 2,\ v = 5$ $$y_1 = a_2 \oplus a_5$$

$$y_0 = a_3 \oplus a_6$$

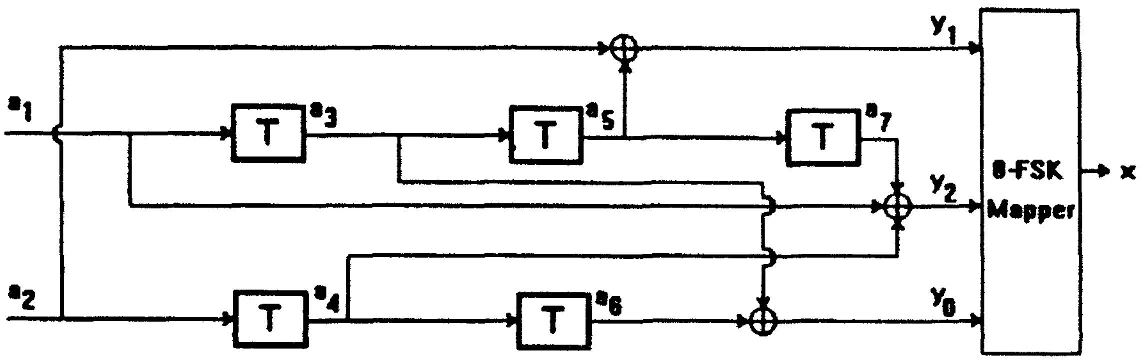


Figure 4.17 TCM encoder for rate 2/3, 8-FSK, 32 state code.

is $d_{free}^2/2E = 3$. The actual d_{free}^2 is $3(2E)$. The asymptotic coding gain of this scheme using (2.3) relative to uncoded 16-FSK modulation which has the same bandwidth efficiency is 1.76 dB.

V. DISCUSSION AND CONCLUSIONS

The objective of this thesis was to investigate trellis coding of M-ary orthogonal signals. While the modulation used here was frequency shift keying, other orthogonal modulation schemes could have been used. The encoder design was presented and resulting encoders were verified for their decodability in the Viterbi algorithm. Several rate 1/2 and rate 2/3 codes were examined, and the following observations surface from those examples. First, it is clear that the asymptotic coding gain, our metric for comparing different codes, strongly depends on the free Euclidean distance of the minimum error event length. In fact, the minimum length, L, is equal to the normalized squared free distance of the code.

As we increase the constraint length, v, the number of states increases exponentially. The corresponding trellis has an increased minimum error event length. In the case of a rate 1/2 code with v=9, we find L=10 and the asymptotic coding gain (ACG) is 3.98 dB. For a rate 2/3 code with v=8, we find L=5 and the ACG is also 3.98 dB.

The general finding of this paper is that compared with uncoded modulation, the same amount of information can be transmitted in the same bandwidth with asymptotic coding gains of 3-4 dB. The areas for further investigation are analysis and simulation in fading channels and using other code rates.

LIST OF REFERENCES

1. Semerciler, Salih, *Coded Orthogonal Signalling*, U. S. Naval Postgraduate School, 1992.

2. Proakis, John G., *Digital Communications*, pp.220-298, McGraw-Hill, New York, 1989.

3. Calderbank, A. Robert, Mazo, J. E., and Wei, Victor, "Asymptotic Upper Bounds on the Minimum Distance of Trellis Codes," *IEEE Transactions on Communications*, vol. COM-33, pp.305-309, Apr. 1985.

4. Ungerboeck, Gottfried, "Trellis-Coded Modulation with Redundant Signal Sets Part 1: Introduction," *IEEE Communications Magazine*, vol. 25, pp. 5-11, Feb. 1987.

5. Bennedetto, Sergio, and others, "Combined Coding and Modulation: Theory and Applications," *IEEE Transactions on Information Theory*, vol. 34, pp.223-236, Mar. 1988.

6. Ungerboeck, Gottfried, "Channel Coding with Multilevel/Phase Signals," *IEEE Transactions on Information Theory*, vol. IT-28, pp.55-67, Jan. 1982.

7. Biglier, Ezio, and others, *Introduction to Trellis-Coded Modulation with Applications*, pp.67-98, Macmillan, New York, 1991.

8. Turgeon, J. Martin, and McLane, Peter J., "Minimal Transmitter Complexity Design of Analytically Described Trellis Codes," *IEEE Transactions on Communications*, vol. 38, pp.1352-1358, Sep. 1990.

9. Forney, G. David, "The Viterbi Algorithm," *Proceedings of the IEEE*, vol. 61, pp.268-278, Mar. 1973.

10. Lin, Shu, and Costello, Daniel, *Error Control Coding: Fundamentals and Applications*, pp.315-346, Prentice-Hall, Inc.

11. Calderbank, Robert, and Mazo, James, "A New Description of Trellis Codes," *IEEE Transactions on Information Theory*, vol. IT-30, pp.784-791, Nov. 1984.

12. Moose, Paul H., *Convolutional Encoder and Soft Viterbi Decoder using Matlab*, Dept. of Electrical and Computer Engineering, Naval Postgraduate School, 1992.

13. Divsalar, Dariush, Simon, Marvin, and Yuen, Joseph, "Trellis Coding with Asymetric Modulations," *IEEE Transactions on Communications*, vol. COM-35, pp.130-141, Feb. 1987.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5000	2
3.	Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Professor Tri T. Ha, Code EC/Ha Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	4
5.	Professor Ralph Hippenstiel, Code EC/Hi Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
6.	CWO2 Michael O. Emerson Rt. 1 Box 526 Hollywood, MD 20636	2
7.	LT Janet E. Stevens 803 Streaker Rd. Sykesville, MD 21784	2